REMARKS/ARGUMENTS

In the Office Action, the Examiner noted that claims 1-13 and 19-39 are pending in the application. The Examiner additionally stated that claims 1-13 and 19-39 are rejected. By this amendment, new claims 40-41 have been added, and claims 1-4, 19, 27, 33, and 39 have been amended. Hence, claims 1-13 and 19-41 are pending in the application.

Applicant hereby requests further examination and reconsideration of the application, in view of the foregoing amendments.

In the Specification

Applicant has amended the specification to secure a substantial correspondence between the claims amended herein and the remainder of the specification. No new matter is presented.

In the Claims

Rejections Under 35 U.S.C. §102(b)

The Examiner rejected claims 1-8, 19-35, and 38-39 under 35 U.S.C. 102(b) as being clearly anticipated by Hoyt et al., U.S. Patent No. 5,604,877 (hereinafter, *Hoyt*). Applicant respectfully traverses the Examiner's rejections.

Hoyt discloses an apparatus for resolving return instructions that includes a branch target buffer (BTB) that includes a return register, and a branch address calculator (BAC) that includes a return stack buffer (RSB). The BTB makes branch predictions at instruction fetch time, and the BAC makes branch predictions at instruction decode time. *Hoyt* maintains two top of stack (TOS) pointers that point to entries in the RSB: a BTB TOS pointer and a BAC TOS pointer. The BTB caches information about previously executed branch instructions. The information includes the type of branch instruction – including whether the branch is a call or return instruction, the target address of the branch instruction, and the outcome of the branch instruction.

When *Hoyt*'s instruction fetch unit fetches an instruction, the BTB predicts whether the instruction is a branch instruction based on the instruction's instruction pointer. If the BTB predicts a call instruction, *Hoyt* performs three actions: 1) increments the BTB TOS

pointer; 2) stores the return address (the address of the instruction following the call instruction) into the return register; and 3) sets a valid bit associated with the return register. *Hoyt* then branches to the target address provided by the BTB. Notably, *Hoyt* does not push the return address onto the RSB when the BTB predicts a call instruction has been fetched, and *Hoyt*'s return register cannot simultaneously store multiple return addresses.

When the call instruction proceeds down the pipeline, *Hoyt*'s instruction decoder decodes the call instruction and notifies the BAC. In response, the BAC increments the BAC TOS pointer and pushes the return address onto the RSB at the BAC TOS pointer.

When the BTB predicts a return instruction has been fetched, *Hoyt* examines the return register valid bit to determine whether the return register is storing a valid return address. If the return register is valid, *Hoyt* branches to the return address stored in the return register; otherwise, *Hoyt* branches to the return address pointed to by the BTB TOS pointer in the RSB. In either case, *Hoyt* clears the return register valid bit and decrements the BTB TOS[1].

When the return instruction proceeds down the pipeline, *Hoyt*'s instruction decoder decodes the call instruction and notifies the BAC. The instruction decoder also passes to the BAC the return address predicted by the BTB at fetch time as described above, i.e., the return address provided by the return register if valid, or by the RSB entry pointed to by the BTB TOS pointer. The BAC pops the return address of the RSB at the BAC TOS pointer and decrements the BAC TOS pointer. The BAC then compares the return address popped off the RSB at the BAC TOS pointer with the return address predicted by the BTB at fetch time. If the two return addresses match, the BAC allows the return instruction to continue down the pipeline for execution. However, if the two return addresses do not match, the BAC flushes the front-end (i.e., the fetch and decode stages) of the pipeline and restarts the front-end using the return address popped off the RSB at the BAC TOS pointer.

[1] The decrementing of the BTB TOS is consistent with TABLE 2 in col. 9 and with the text throughout (e.g., col. 11, lines 1-4, 30-33; col. 12, lines 27-29). However, Applicant notes Fig. 6 appears to have a typographical error, namely the line "BAC TOC ← BAC TOS-1 (If the BTB predicted the Return)" apparently should be "~~BAC~~ BTB TOC ← ~~BAC~~ BTB TOS-1 (If the BTB predicted the Return)."

Applicant notes that because the RSB is a single memory shared by the BTB and BAC, *Hoyt* does not push a return address onto the RSB when a call instruction is fetched and also push a return address onto the RSB when a call instruction is decoded; rather, *Hoyt* only pushes a return address onto the RSB when a call instruction is decoded, but does not when a call instruction is fetched. Instead, when predicting return instructions, the BTB must rely on the BAC to have pushed the return address onto the RSB. This fact, coupled with the fact that *Hoyt*'s return register is only capable of storing a single return address at a time, rather than being capable of storing multiple return addresses from multiple call instructions simultaneously, would cause a microprocessor employing *Hoyt*'s return address prediction mechanism to mispredict the return address in the following situation.

Assume a program calls a first subroutine which calls a second subroutine which returns and then the first subroutine returns. This common situation would generate an instruction sequence comprising a call instruction followed by another call instruction followed by a return instruction followed by a return instruction, each of which may be interspersed with other non-call/return instructions. When the second return instruction is fetched, the return register will be invalid, causing the microprocessor to branch to the return address predicted by the RSB at the BTB TOS pointer. However, if by this time the first call instruction has not reached the instruction decoder to cause the correct return address to be pushed onto the RSB, then the return address predicted by the RSB at the BTB TOS pointer will be incorrect. This condition may occur if the instruction fetcher is fetching instructions sufficiently faster than the instruction decoder is decoding them to keep an instruction buffer between the instruction fetcher and the instruction decoder sufficiently full to prevent the first call instruction from reaching the instruction decoder before the second return instruction is fetched. In contrast, Applicant's invention is advantageously capable of accurately predicting the return address in such a situation since the first and second call/return stacks are distinct, the first call/return stack is capable of storing multiple return addresses pushed onto the first call/return stack when a call instruction is fetched.

With respect to amended claim 1, Applicant respectfully asserts that *Hoyt* does not teach a call/return stack, that simultaneously stores a plurality of return addresses, configured for pushing thereon a plurality of return addresses in response to fetching from an instruction cache a plurality of cache lines predicted to include a corresponding plurality of call instructions, wherein each of the return addresses is pushed onto the first call/return stack prior to decoding the corresponding call instruction, which are limitations recited in amended claim 1. Although *Hoyt* loads his return register prior to decoding a call instruction, the return register is only capable of simultaneously storing a single return address; and *Hoyt* only pushes return addresses onto his RSB when he decodes a call instruction. For these reasons, Applicant respectfully asserts that *Hoyt* does not anticipate amended claim 1.

With respect to claims 2-13, these claims depend from claim 1 and add further limitations that are not anticipated by *Hoyt*. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejections to claims 2-13.

With respect to amended claim 19, Applicant respectfully asserts that *Hoyt* does not teach a call/return stack that simultaneously stores a plurality of return addresses, wherein the return addresses are pushed onto the first call/return stack in response to indications of previously executed call instructions provided from a BTAC in response to an instruction cache fetch address, which are limitations recited in amended claim 19. Although *Hoyt* loads his return register in response to the BTB indicating a call instruction has been fetched, the return register is only capable of simultaneously storing a single return address; and *Hoyt* only pushes return addresses onto his RSB when he decodes a call instruction, rather than in response to the BTB indicating a call instruction has been fetched. For these reasons, Applicant respectfully asserts that *Hoyt* does not anticipate amended claim 19.

With respect to claims 20-26, these claims depend from claim 19 and add further limitations that are not anticipated by *Hoyt*. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejections to claims 20-26.

With respect to amended claim 27, Applicant respectfully asserts that *Hoyt* does not teach pushing onto a first call/return stack a plurality of return addresses of a corresponding plurality of call instructions causing the return addresses to be simultaneously stored in the first call/return stack, where for each of the return addresses the pushing is performed prior to decoding of the corresponding call instruction, which are limitations recited in amended claim 27. Although *Hoyt* loads his return register prior to decoding a call instruction, the return register is only capable of simultaneously storing a single return address; and *Hoyt* only pushes return addresses onto his RSB when he decodes a call instruction. For these reasons, Applicant respectfully asserts that *Hoyt* does not anticipate amended claim 27.

With respect to claims 28-38, these claims depend from claim 27 and add further limitations that are not anticipated by *Hoyt*. Accordingly, Applicant respectfully requests that the Examiner withdraw his rejections to claims 28-38.

With respect to amended claim 39, Applicant respectfully asserts that *Hoyt* does not teach a speculative call/return stack that simultaneously stores a plurality of return addresses that are pushed onto the speculative call/return stack in response to instances of a speculative BTAC indicating a speculative presence of a call instruction in a line of instructions fetched from an instruction cache, which are limitations recited in amended claim 39. Although *Hoyt* loads his return register in response to the BTB indicating a call instruction has been fetched, the return register is only capable of simultaneously storing a single return address; and *Hoyt* only pushes return addresses onto his RSB when he decodes a call instruction, rather than in response to the BTB indicating a call instruction has been fetched. For these reasons, Applicant respectfully asserts that *Hoyt* does not anticipate amended claim 39.

With respect to new claims 40 and 41, Applicant respectfully asserts that *Hoyt* does not anticipate new claims 40 and 41 for similar reasons discussed above.

CONCLUSIONS

In view of the arguments advanced above, Applicant respectfully submits that claims 1-13 and 19-41 are in condition for allowance. Reconsideration of the rejections is requested, and allowance of the claims is solicited.

Applicant earnestly requests that the Examiner contact the undersigned practitioner by telephone if the Examiner has any questions or suggestions concerning this amendment, the application, or allowance of any claims thereof.

EXPRESS MAIL LABEL NUMBER: EO 006 076 585 US
DATE OF DEPOSIT: 10/6/05
I hereby certify that this paper is being deposited with the U.S. Postal Service Express Mail Post Office to Addressee Service under 37 C.F.R. §1.10 on the date shown above and is addressed to Mail Stop , Commissioner for Patents, PO Box 1450, Alexandria, VA 22313-1450.
Lucia Merion

Respectfully submitted,

/E. Alan Davis/
By: _____

E. ALAN DAVIS
Registration No. 39,954
Tel: (512) 301-7234

10/6/2005
Date: _____